Data Communication and Net-Centric Computing

COSC 1111/2061/1110

Lecture 1

Introduction

Course Details

- Lecturer: Dr. Abdun Mahmood
  - Room: 14.11.31
  - Email: abdun.mahmood@rmit.edu.au
  - Phone: 9925 1902
  - Consultation hours: 2:30PM-4:30PM on Thursday
- Course Code
  - Undergraduate: COSC1111
  - Postgraduate: COSC2061
- Head Tutor: Writabrata Banerjee (E73367@rmit.edu.au)

Lecture Overview

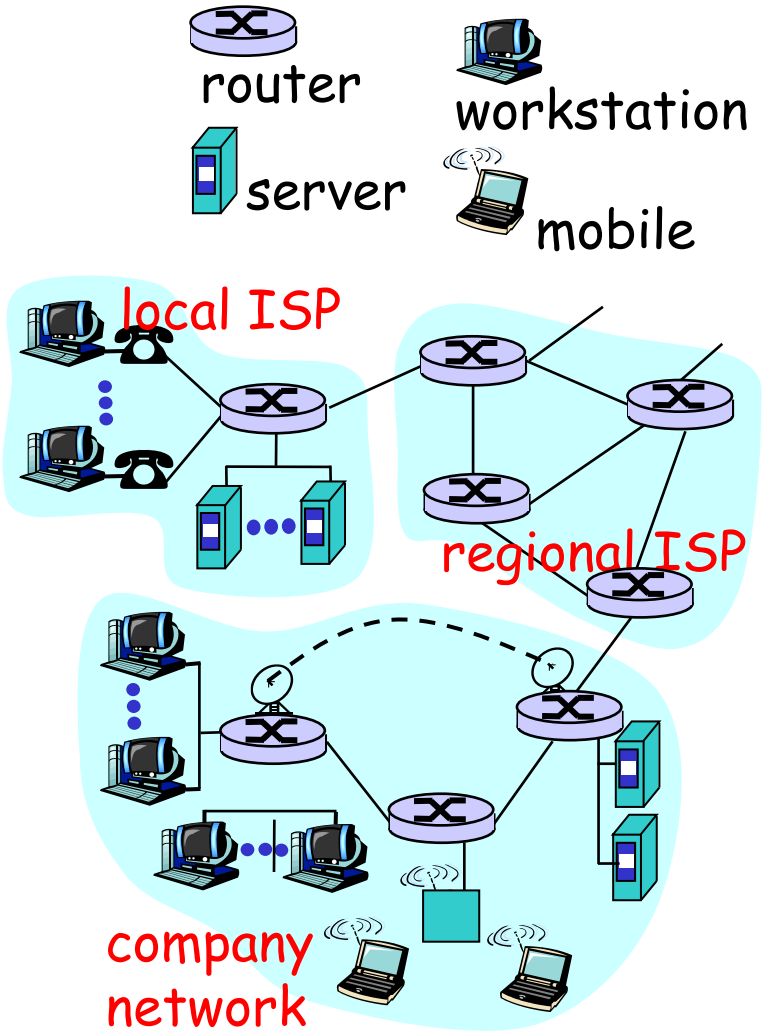
- During this lecture, we will understand
  - what's the Internet
  - what's a protocol?
  - network edge, network core
  - access net, physical media
  - Internet/ISP structure
  - performance: loss, delay
  - protocol layers, service models

- Recommended reading
  - Chapter 1,2 (Stallings)

What's the Internet: "nuts and bolts" view

- millions of connected computing devices: *hosts, end-systems*
  - PCs workstations, servers
  - PDAs phones, toasters

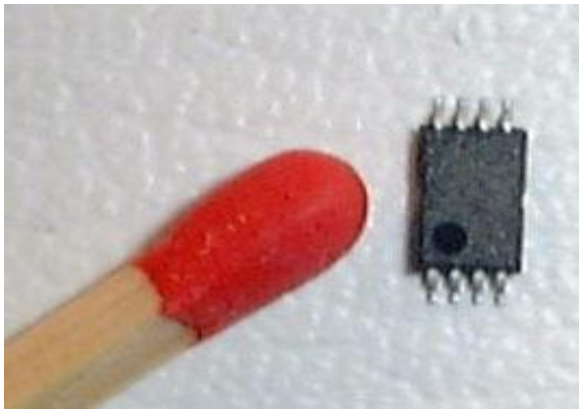
  running *network apps*
- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate = *bandwidth*
- *routers:* forward packets (chunks of data)



“Cool” internet appliances

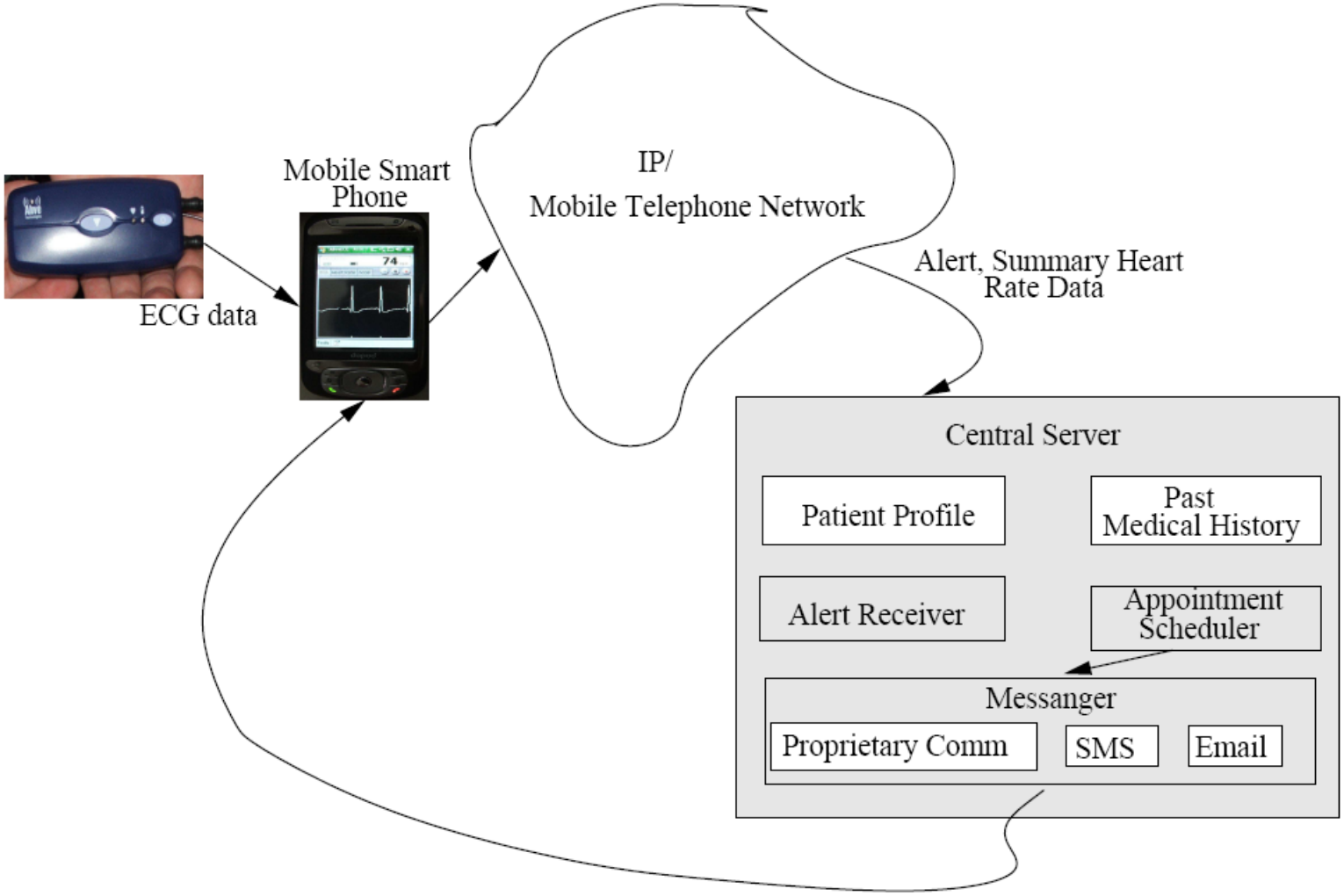
IP picture frame
http://www.ceiva.com/

Web-enabled toaster+weather forecaster

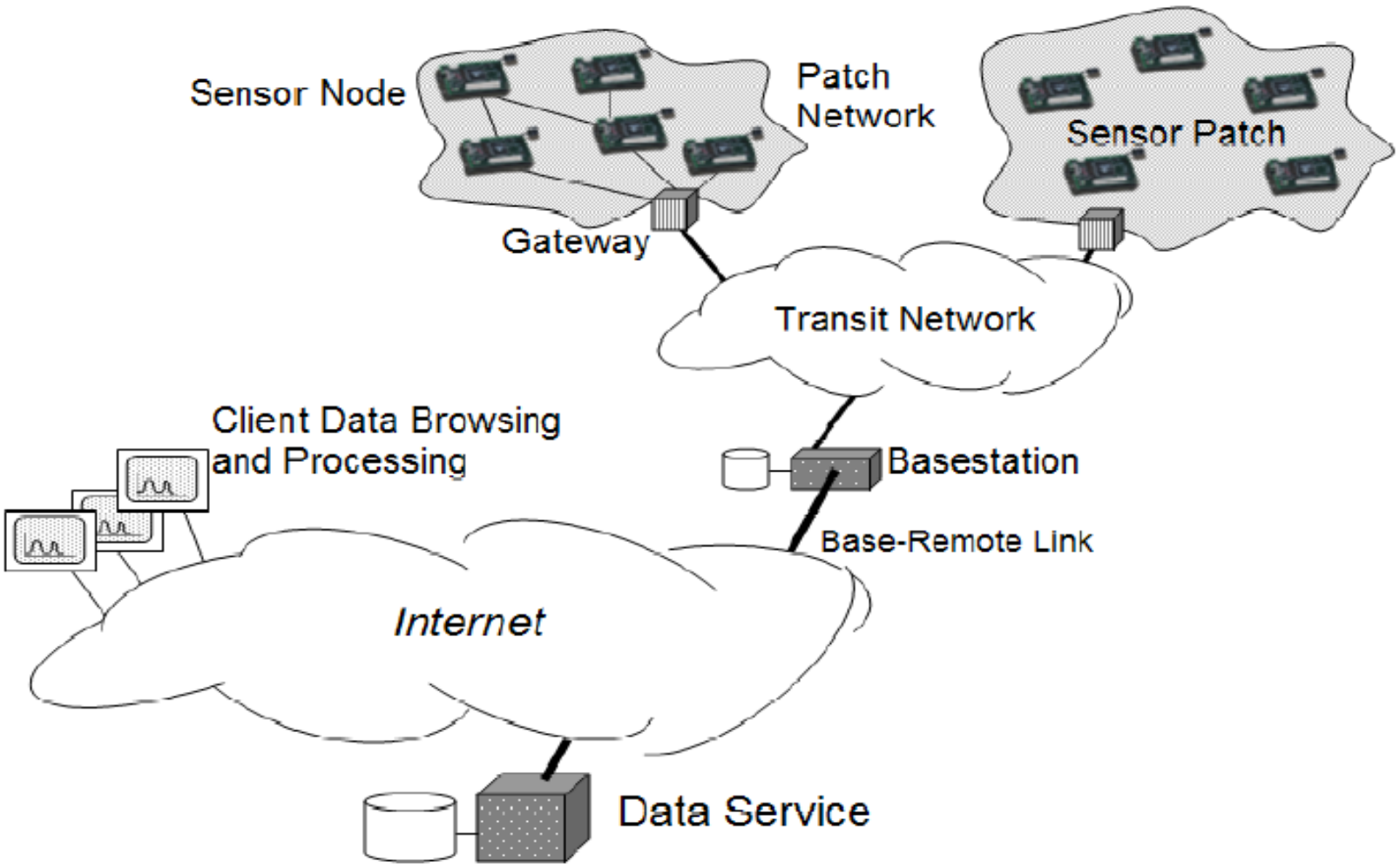


World’s *smallest* web server
http://d116.com/ace/images/ACE-match-topview.jpg

“Cool” internet appliances- Realtime ECG monitoring

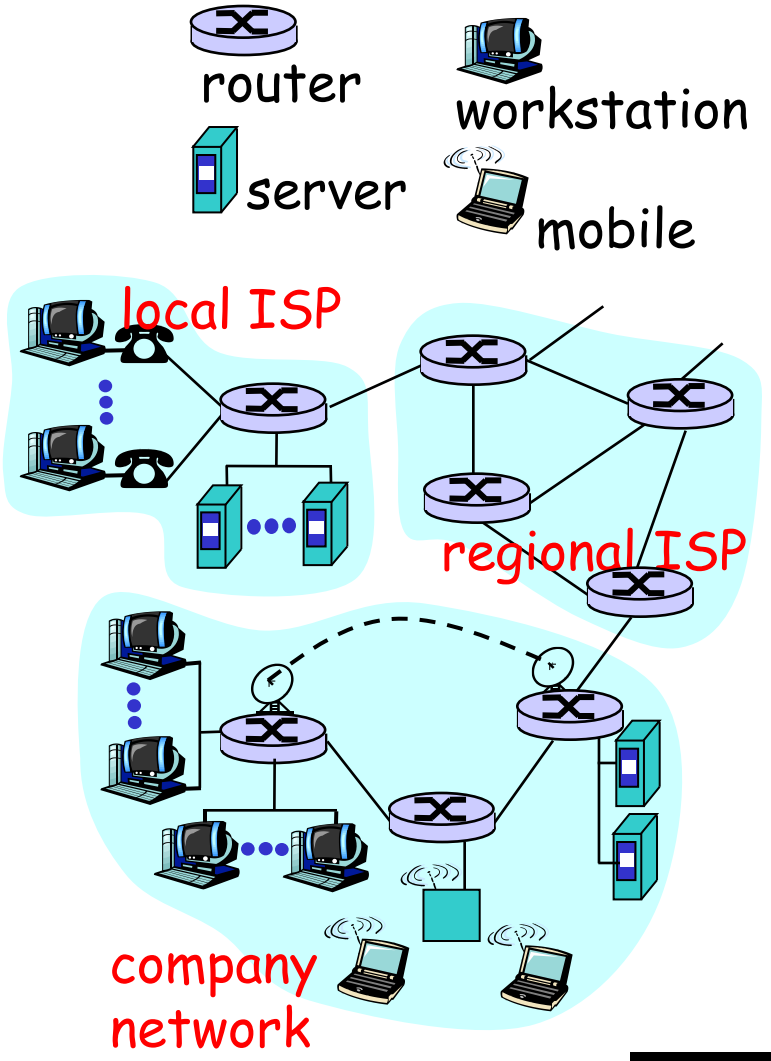


“Cool” internet appliances- Realtime monitoring of environment/building etc.



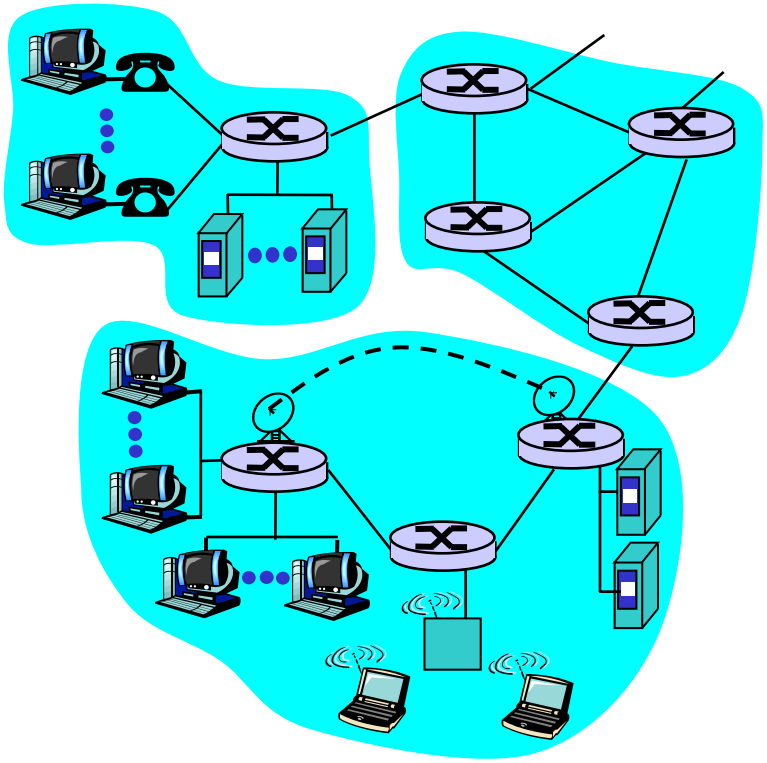
What's the Internet: "nuts and bolts" view

- *protocols* control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, FTP, PPP
- *Internet:* "network of networks"
  - loosely hierarchical
  - public Internet versus private intranet
- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



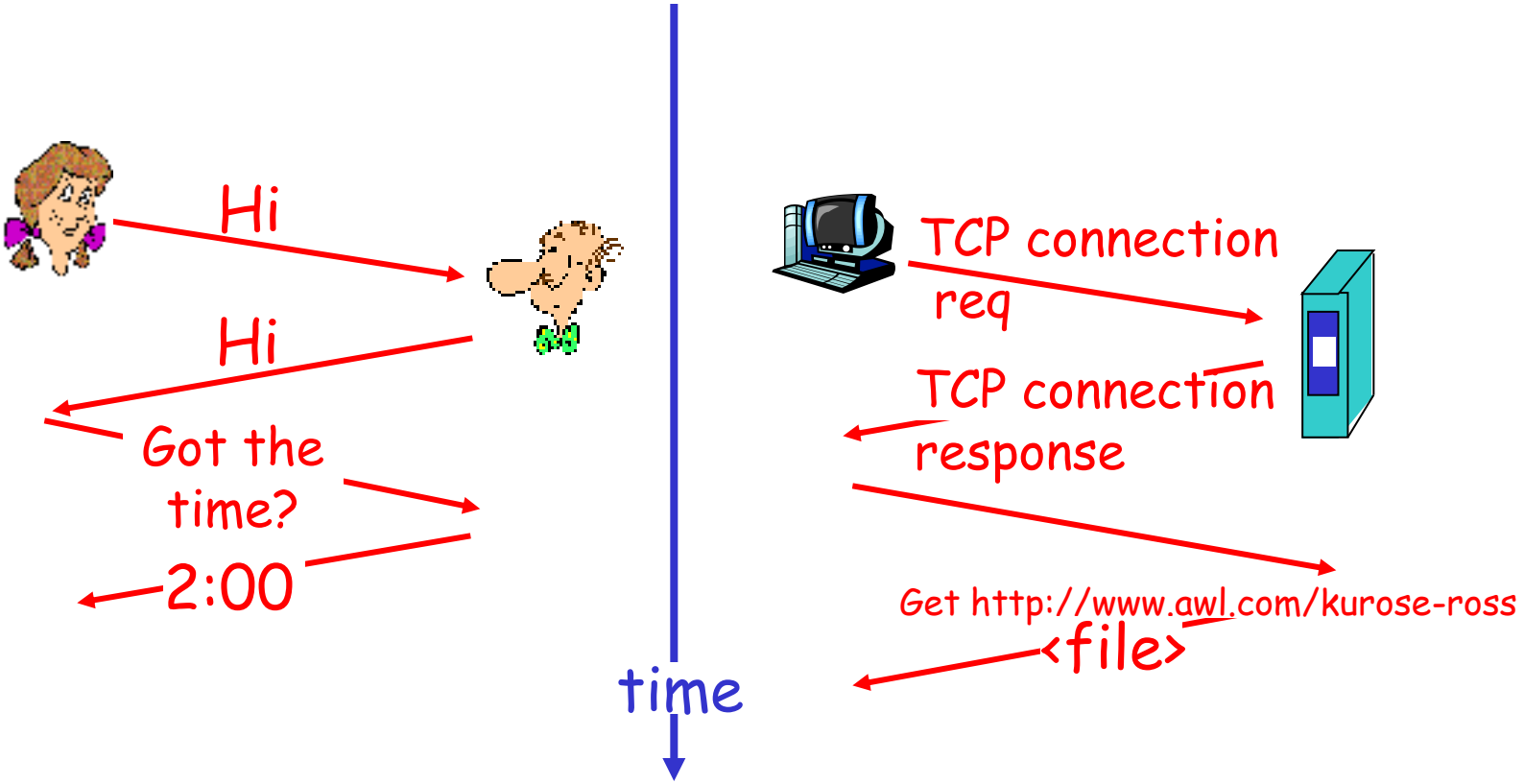
What's the Internet: a service view

- communication *infrastructure* enables distributed applications:
  - Web, email, games, e-commerce, database., voting, file (MP3) sharing
- communication services provided to apps:
  - connectionless
  - connection-oriented



What's a protocol?

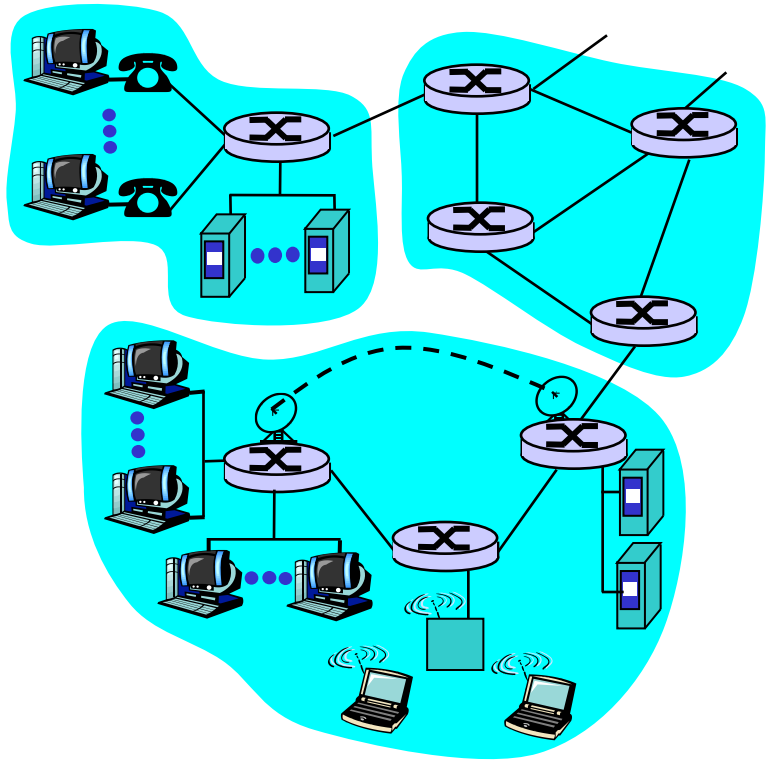
a human protocol and a computer network protocol:



- ***protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt***
- ***all communication activity in Internet governed by protocols***

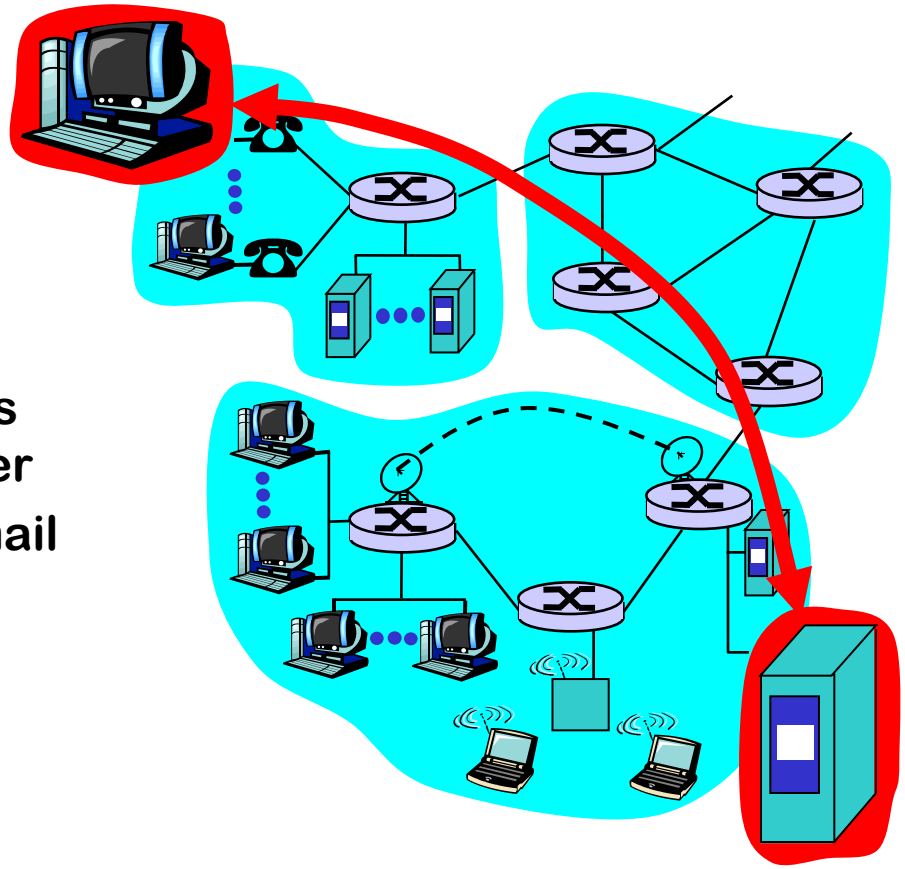
A closer look at network structure:

- **network edge:** applications and hosts
- **network core:**
  - routers
  - network of networks
- **access networks, physical media:** communication links



The network edge:

- **end systems (hosts):**
  - run application programs
  - e.g. Web, email
  - at "edge of network"
- **client/server model**
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server
- **peer-peer model:**
  - minimal (or no) use of dedicated servers
  - e.g. Gnutella, KaZaA



Network edge: connection-oriented service

Goal: data transfer between end systems

- *handshaking:* setup (prepare for) data transfer ahead of time
  - Hello, hello back human protocol
  - *set up "state"* in two communicating hosts
- TCP - Transmission Control Protocol
  - Internet's connection-oriented service

TCP service [RFC 793]

- *reliable, in-order* byte-stream data transfer
  - loss: acknowledgements and retransmissions
- *flow control:*
  - sender won't overwhelm receiver
- *congestion control:*
  - senders "slow down sending rate" when network congested

Network edge: connectionless service

Goal: data transfer between end systems

- same as before!

- **UDP** - User Datagram Protocol [RFC 768]: Internet's connectionless service
  - unreliable data transfer
  - no flow control
  - no congestion control

App's using TCP:

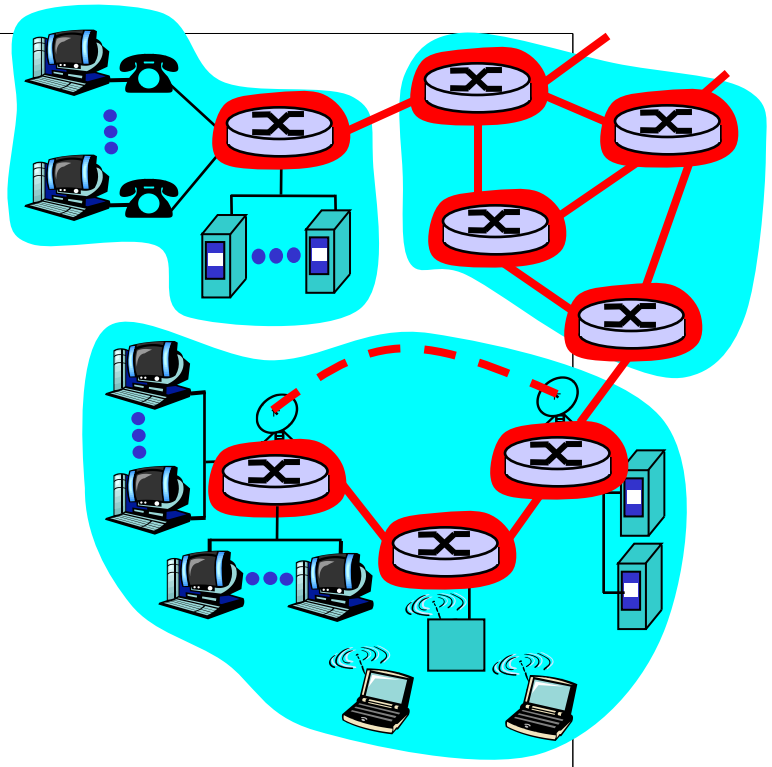
- HTTP (Web), FTP (file transfer), Telnet (remote login), SMTP (email)

App's using UDP:

- streaming media, teleconferencing, DNS, Internet telephony

The Network Core

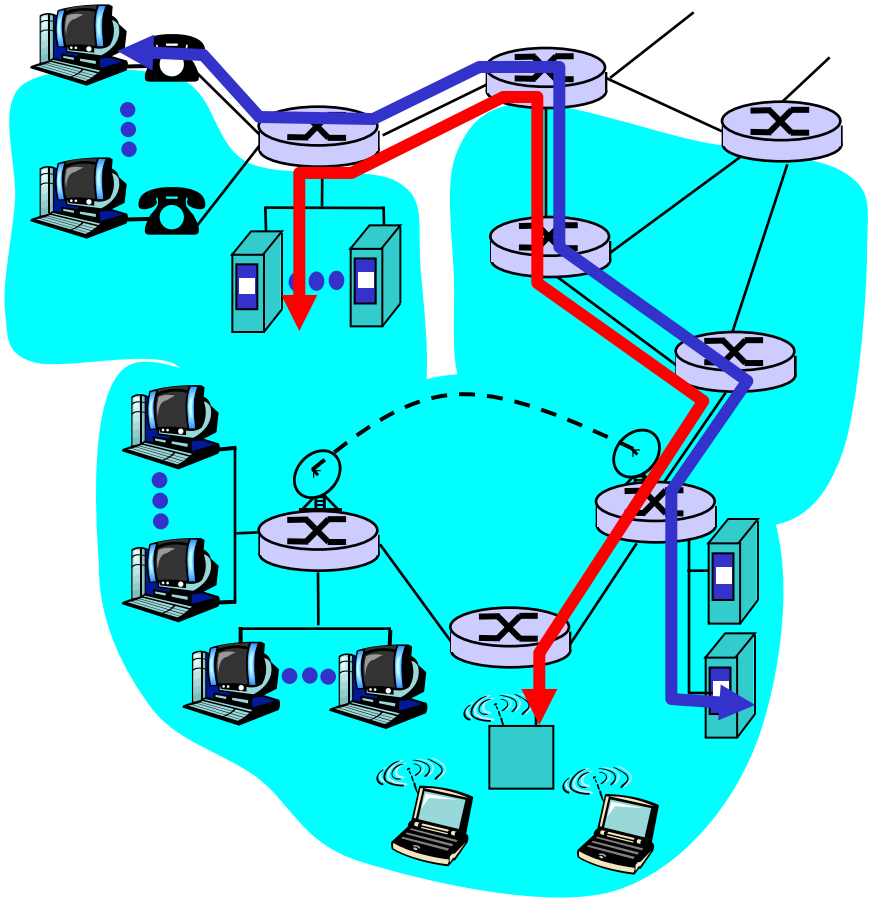
- mesh of interconnected routers
- *the* fundamental question: how is data transferred through net?
  - circuit switching: dedicated circuit per call: telephone net
  - packet-switching: data sent thru net in discrete "chunks"



Network Core: Circuit Switching

End-end resources reserved for "call"

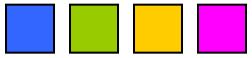
- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required
- dividing link bandwidth into "pieces"
  - frequency division
  - time division



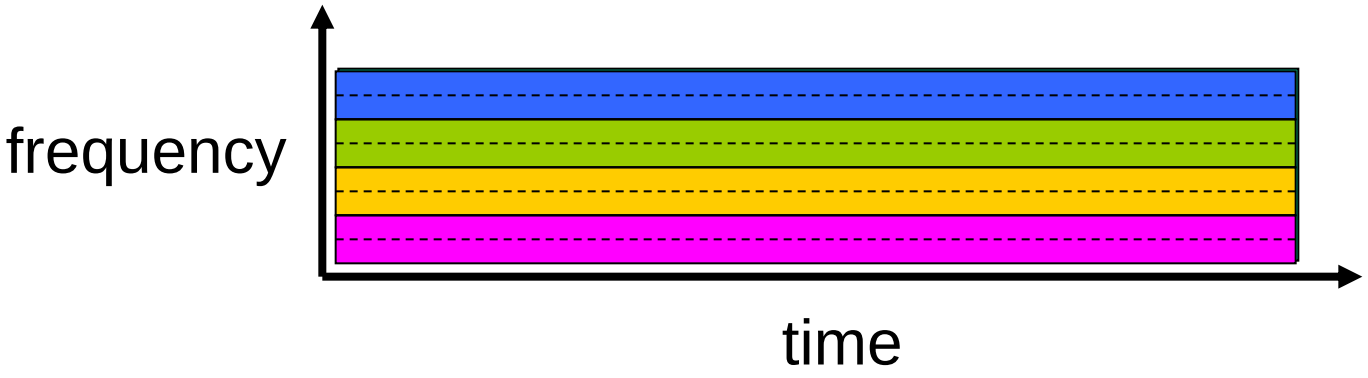
Circuit Switching: FDMA and TDMA

Example:

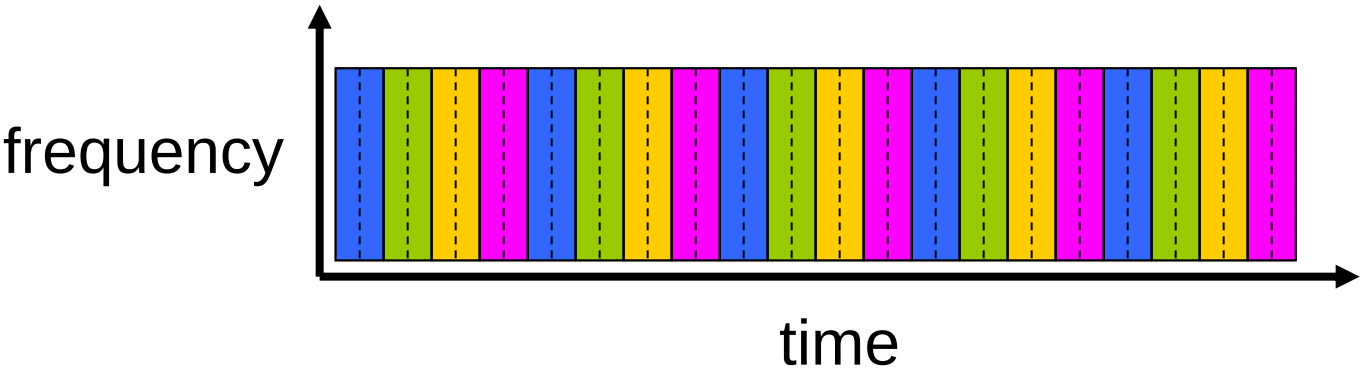
4 users



FDMA



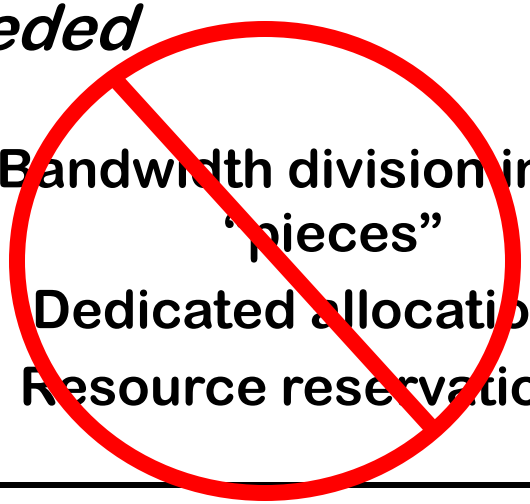
TDMA



Network Core: Packet Switching

each end-end data stream divided into *packets*

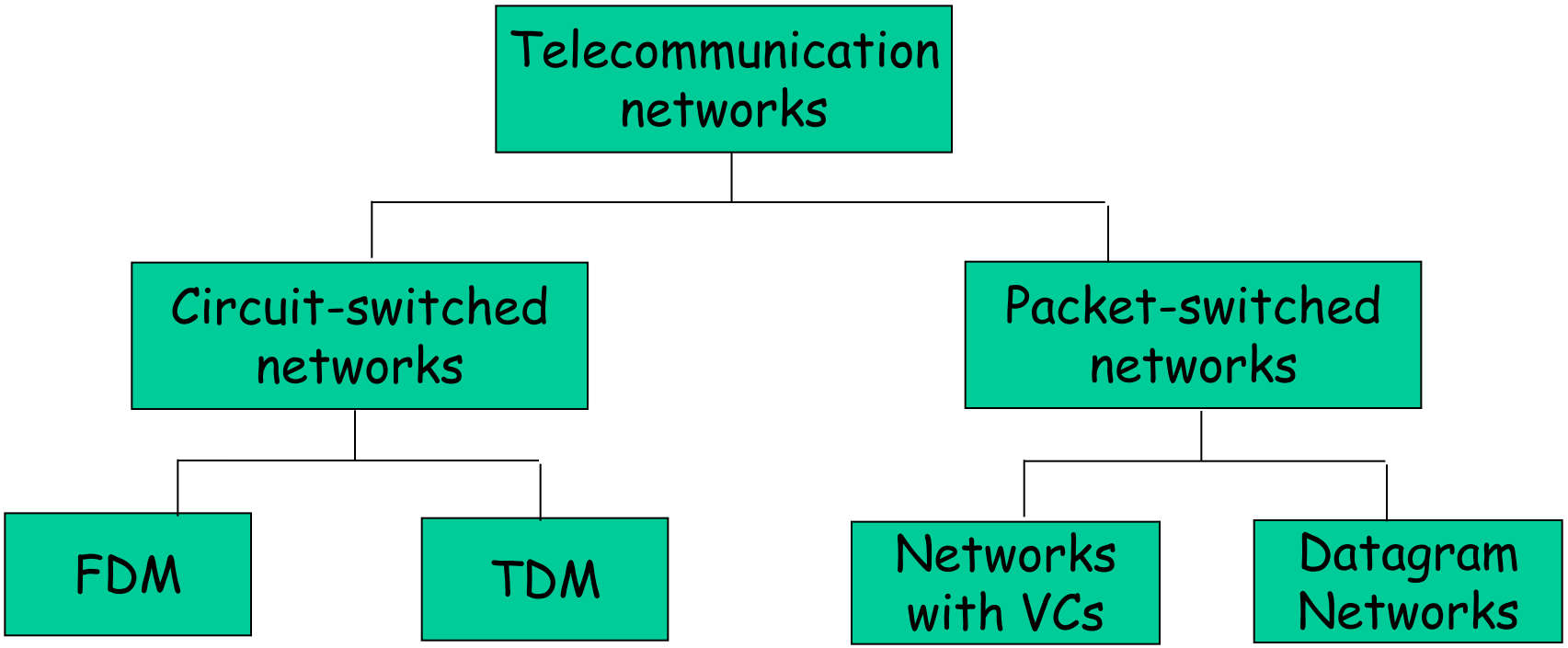
- user A, B packets *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*



resource contention:

- aggregate resource demand can exceed amount available
- congestion: packets queue, wait for link use
- store and forward: packets move one hop at a time
  - transmit over link
  - wait turn at next link

Network Taxonomy



- Datagram network is *not* either connection-oriented or connectionless.
- Internet provides both connection-oriented (TCP) and connectionless services (UDP) to apps.

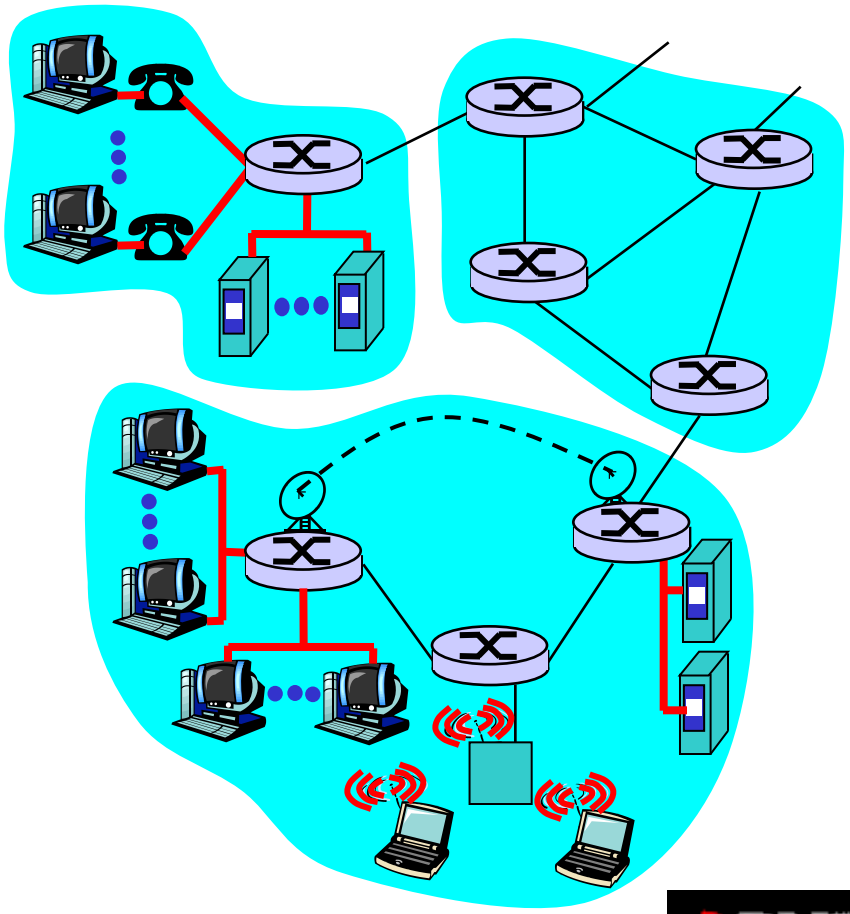
Access networks and physical media

Q: How to connect end systems to edge router?

- residential access nets
- institutional access networks (school, company)
- mobile access networks

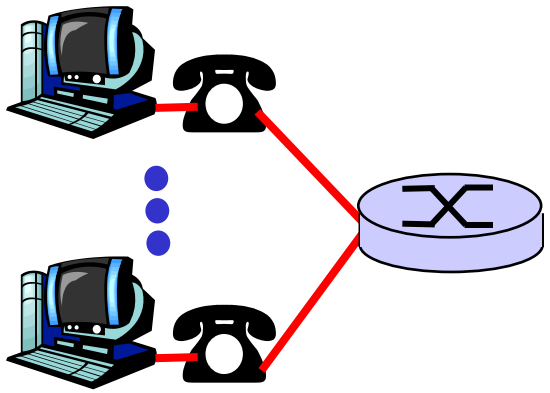
Keep in mind:

- bandwidth (bits per second) of access network?
- shared or dedicated?



Residential access: point to point access

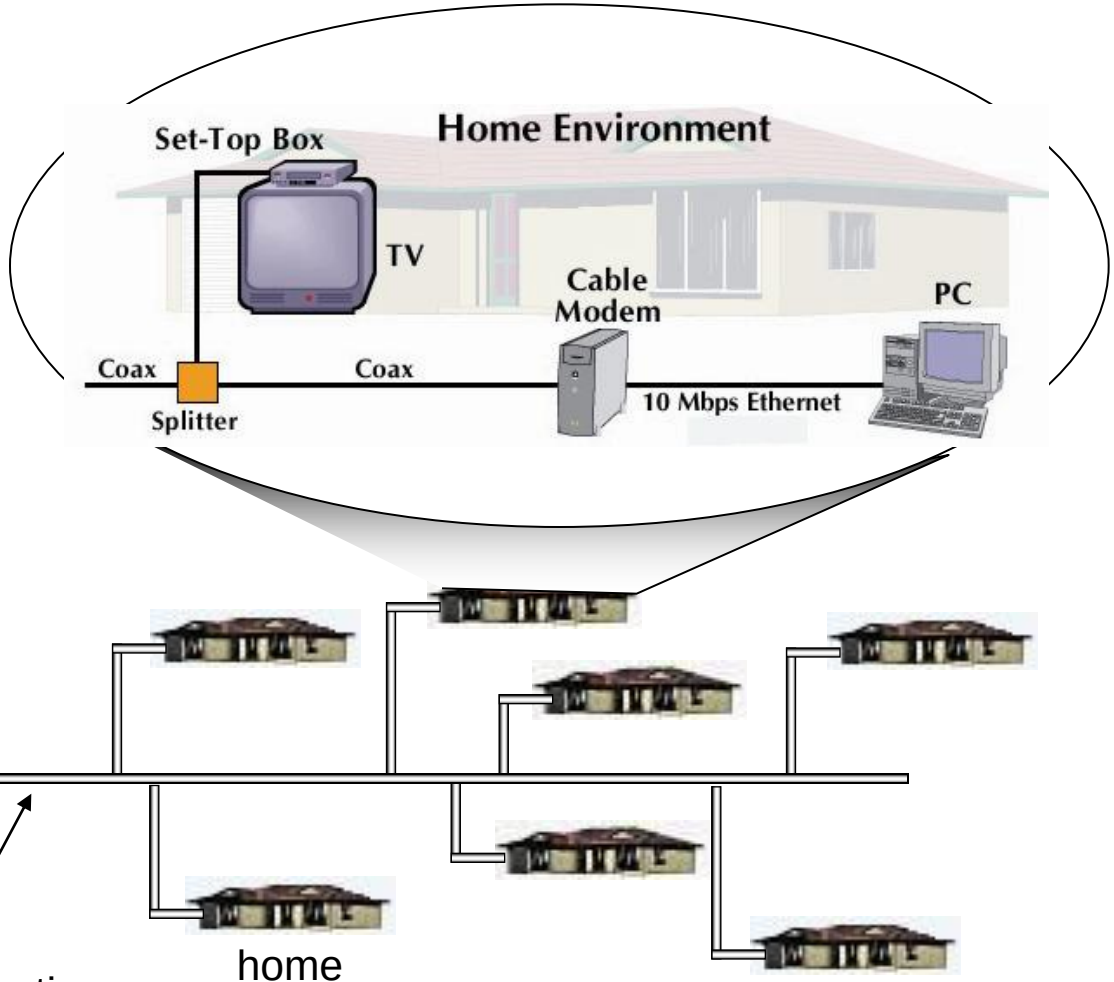
- Dialup via modem
  - up to 56Kbps direct access to router (often less)
  - Can't surf and phone at same time: can't be "always on"



- ADSL: asymmetric digital subscriber line
  - up to 1 Mbps upstream (today typically < 256 kbps)
  - up to 8 Mbps downstream (today typically < 1 Mbps)
  - FDM: 50 kHz - 1 MHz for downstream
    4 kHz - 50 kHz for upstream
    0 kHz - 4 kHz for ordinary telephone

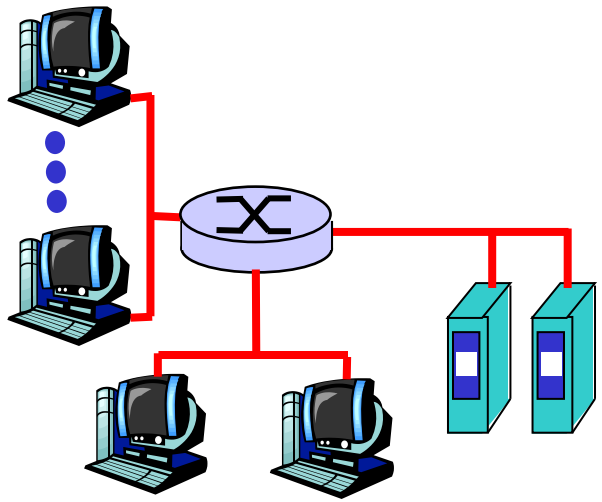
- network of cable and fiber attaches homes to ISP router

Cable Network Architecture: Overview



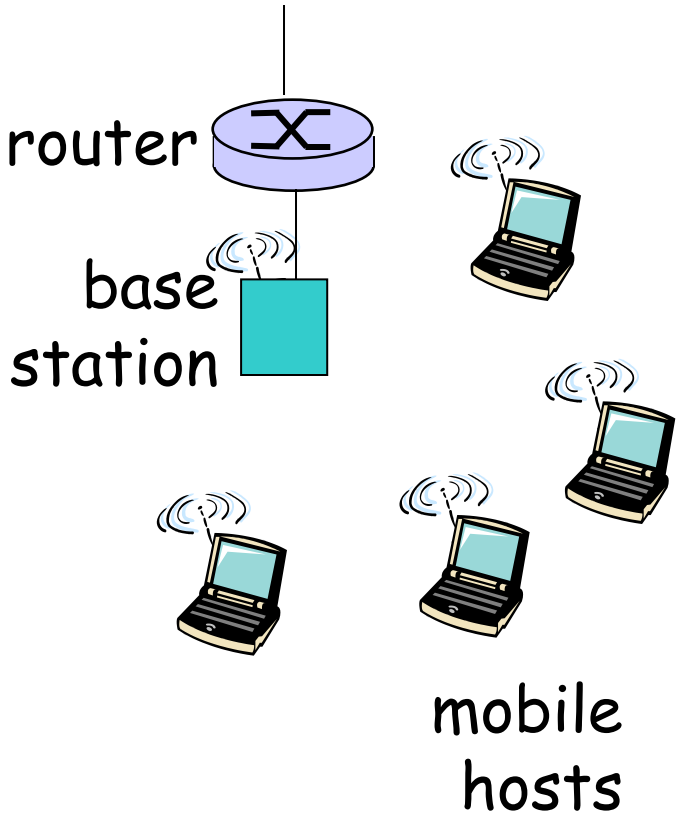
Company access: local area networks

- company/univ **local area network** (LAN) connects end system to edge router
- **Ethernet:**
  - shared or dedicated link connects end system and router
  - 10 Mbs, 100Mbps, Gigabit Ethernet
- **deployment:** institutions, home LANs happening now



Wireless access networks

- shared *wireless* access network connects end system to router
  - via base station aka “access point”
- wireless LANs:
  - 802.11 (WiFi): 11 Mbps-54Mbps
- wider-area wireless access
  - provided by telco operator
  - 3G ~ 384 kbps
  - WAP/GPRS



Physical Media

- **Bit:** propagates between transmitter/rcvr pairs
- **physical link:** what lies between transmitter & receiver
- **guided media:**
  - signals propagate in solid media: copper, fiber, coax
- **unguided media:**
  - signals propagate freely, e.g., radio

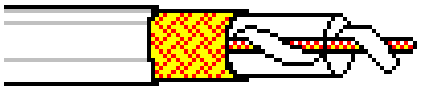
Twisted Pair (TP)

- two insulated copper wires
  - Category 3: traditional phone wires, 10 Mbps Ethernet
  - Category 5 TP: 100Mbps Ethernet

Physical Media: coax, fiber

Coaxial cable:

- two concentric copper conductors
- bidirectional
- baseband:
  - single channel on cable
  - legacy Ethernet
- broadband:
  - multiple channel on cable
  - HFC



Fiber optic cable:

- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (e.g., 5 Gps)
- low error rate: repeaters spaced far apart ; immune to electromagnetic noise

Physical media: radio

- signal carried in electromagnetic spectrum
- no physical "wire"
- bidirectional
- propagation environment effects:
  - reflection
  - obstruction by objects
  - interference

Radio link types:

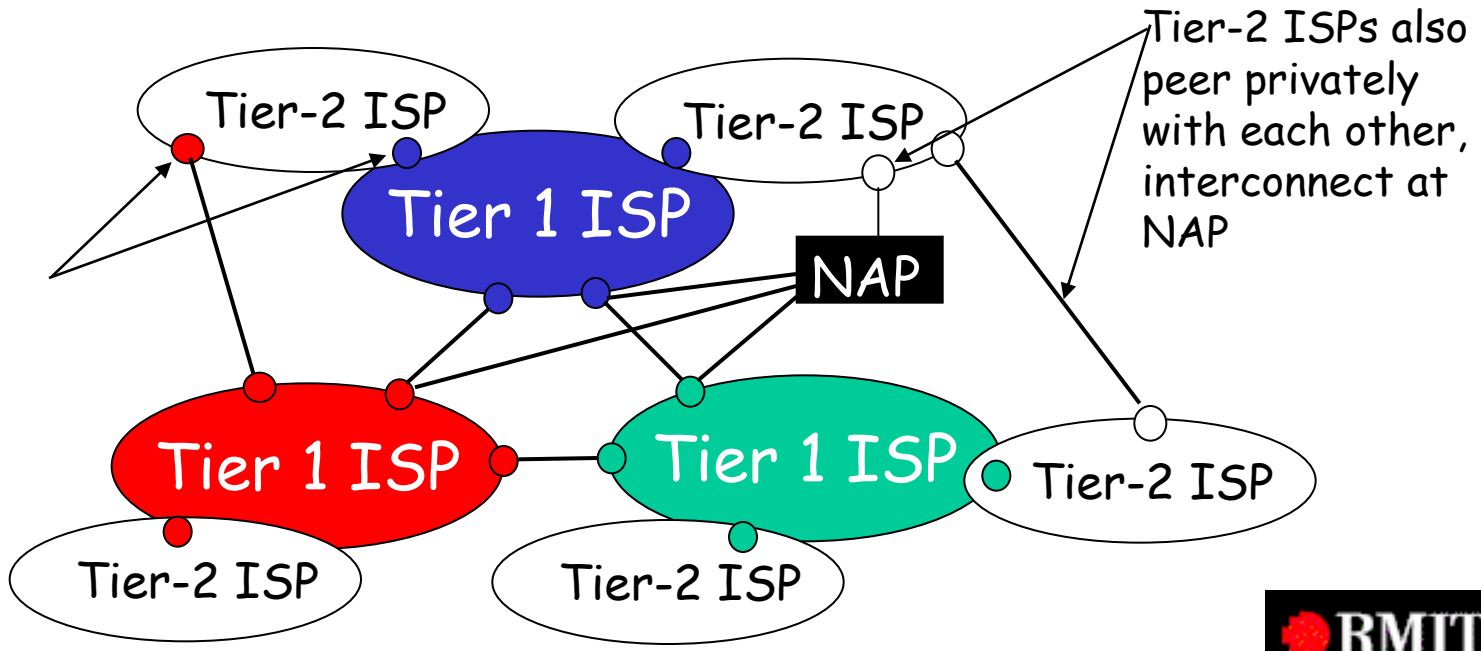
- terrestrial microwave
  - e.g. up to 45 Mbps channels
- LAN (e.g., WaveLAN)
  - 2Mbps, 11Mbps
- wide-area (e.g., cellular)
  - e.g. 3G: hundreds of kbps
- satellite
  - up to 50Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus LEOS

Internet structure: network of networks

- roughly hierarchical
- at center: "tier-1" ISPs (e.g., UUNet, BBN/Genuity, Sprint, AT&T), national/international coverage
  - treat each other as equals

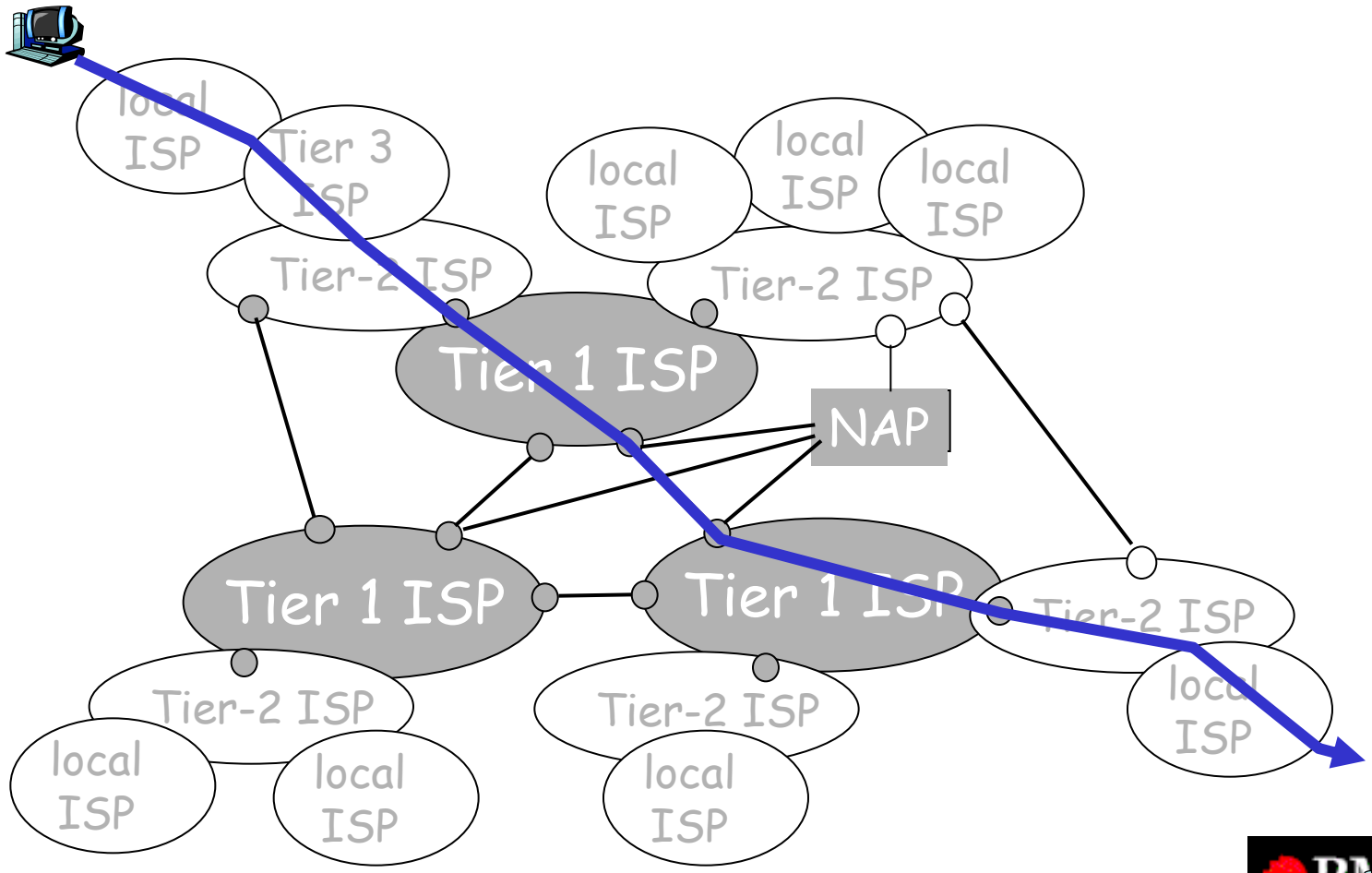
- "Tier-2" ISPs: smaller (often regional) ISPs
  - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs
- "Tier-3" ISPs and local ISPs
  - last hop ("access") network (closest to end systems)

Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet
- tier-2 ISP is *customer* of tier-1 provider



Internet structure: network of networks

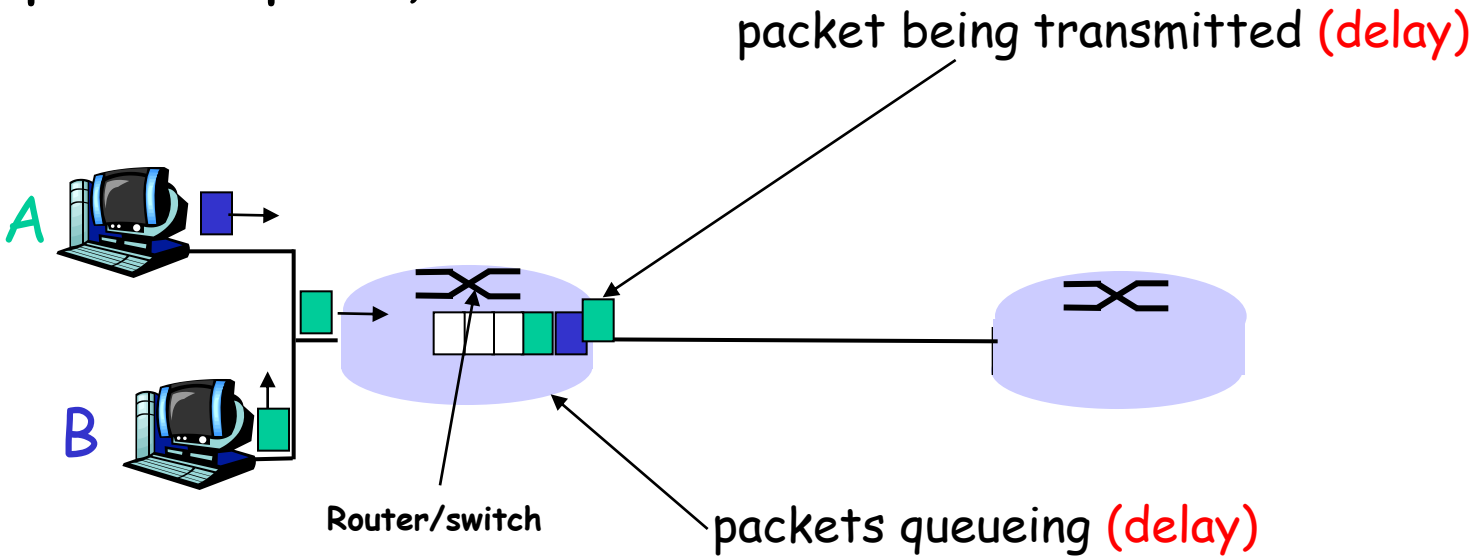
- a packet passes through many networks!



How do loss and delay occur?

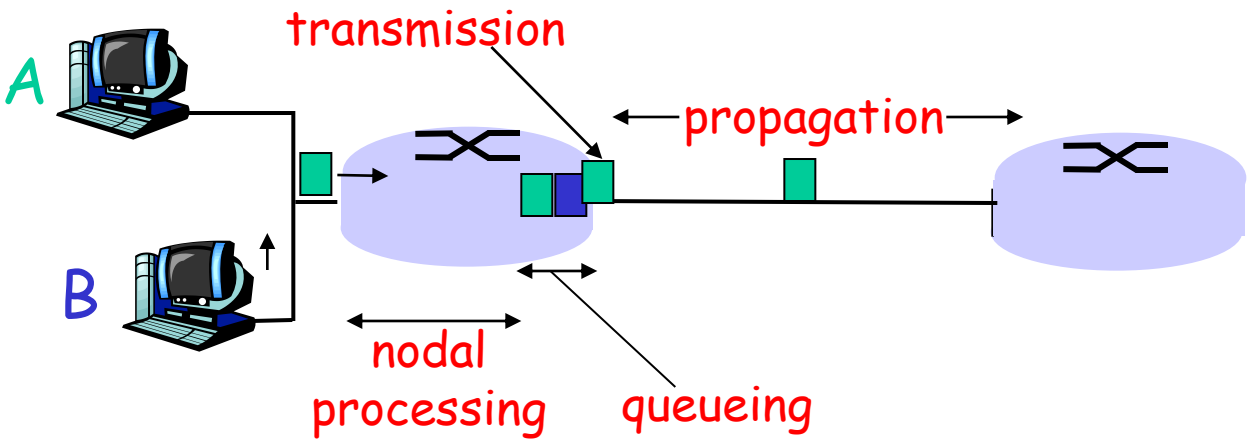
packets *queue* in router buffers

- packet arrival rate to link exceeds output link capacity
- packets queue, wait for turn



- free (available) buffers: arriving packets dropped (loss) if no free buffers
- lost packet may be retransmitted by previous node, by source end system, or not retransmitted at all

Four sources of packet delay



The delays are:

- nodal processing (check bit errors)
- Queueing (time waiting at output link for transmission)
- Transmission (time required to push bits from buffer)
- Propagation (distance/speed)

Protocol "Layers"

Networks are complex!

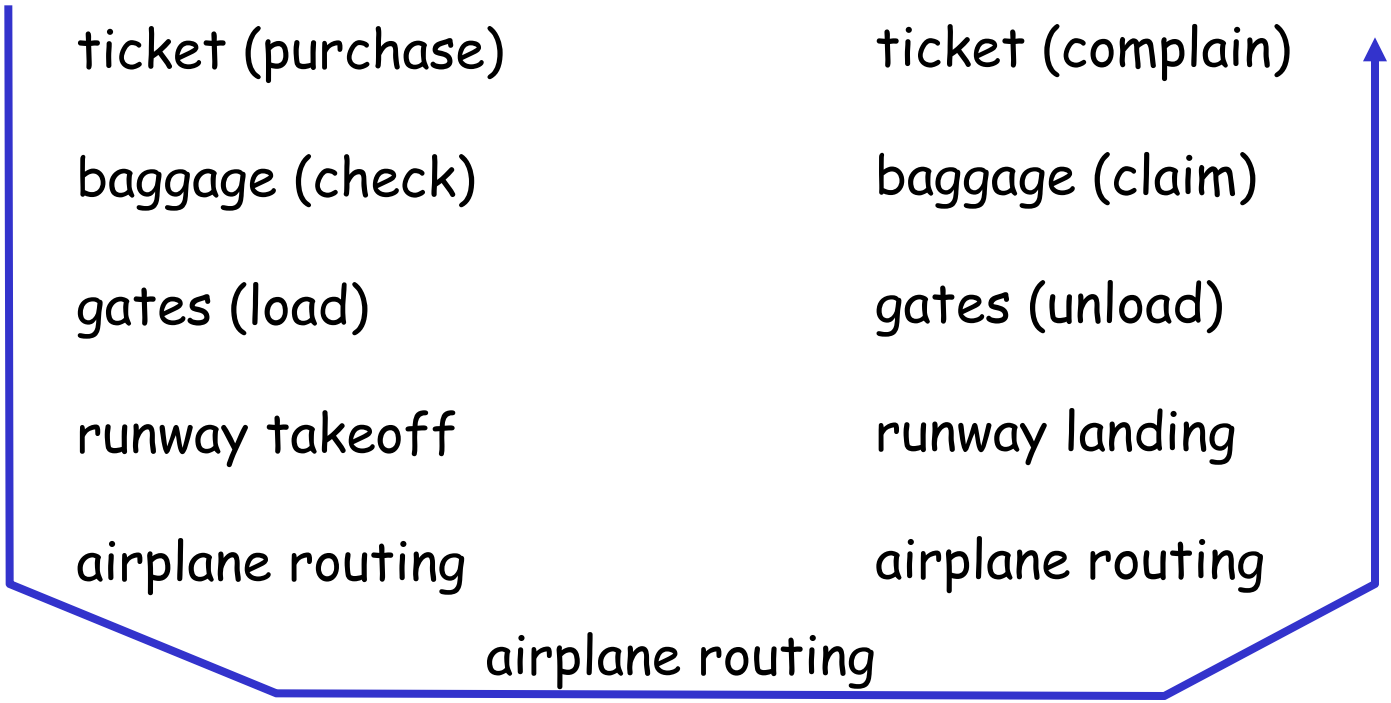
- many "pieces":
  - hosts
  - routers
  - links of various media
  - applications
  - protocols
  - hardware, software

Question:

Is there any hope of *organizing* structure of network?

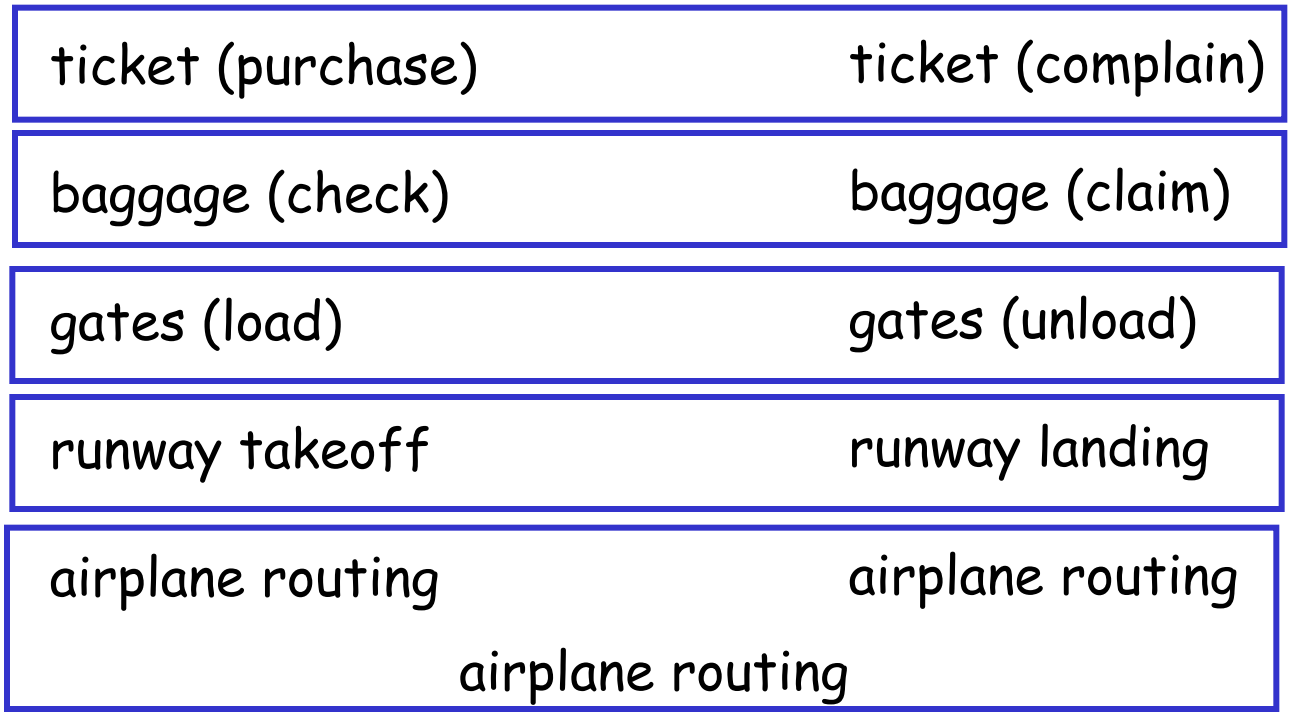
Or at least our discussion of networks?

Organization of air travel



- a series of steps

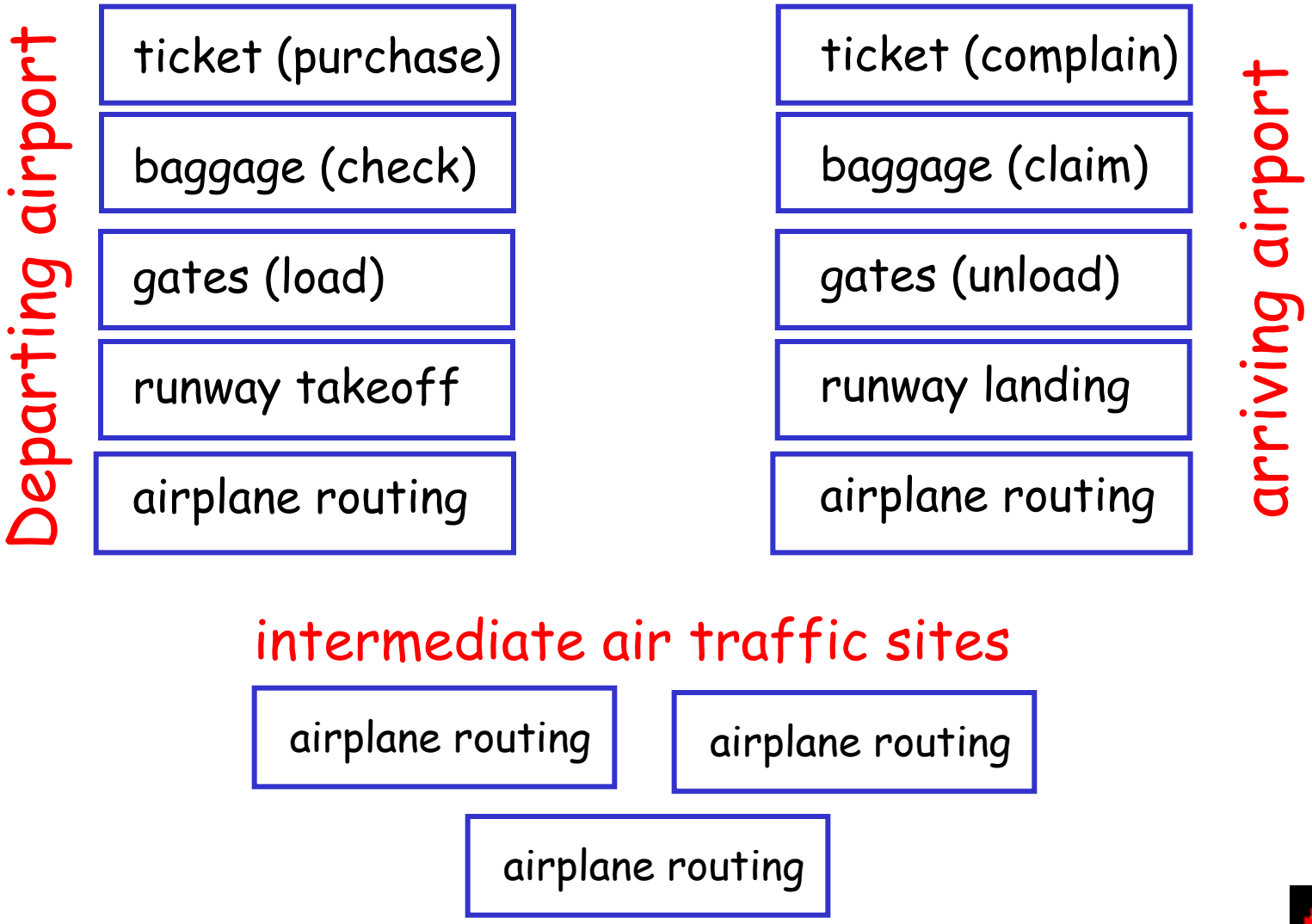
Organization of air travel: a different view



Layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

Distributed implementation of layer functionality



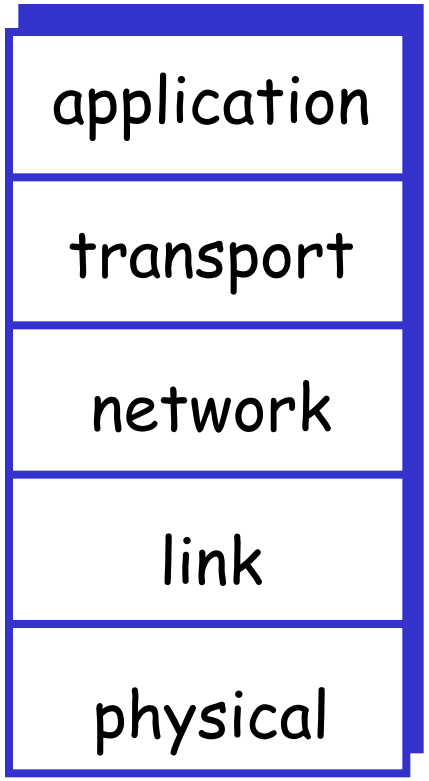
Why layering?

Dealing with complex systems:

- explicit structure allows identification, relationship of complex system's pieces
  - layered reference model for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- layering considered harmful?

Internet protocol stack

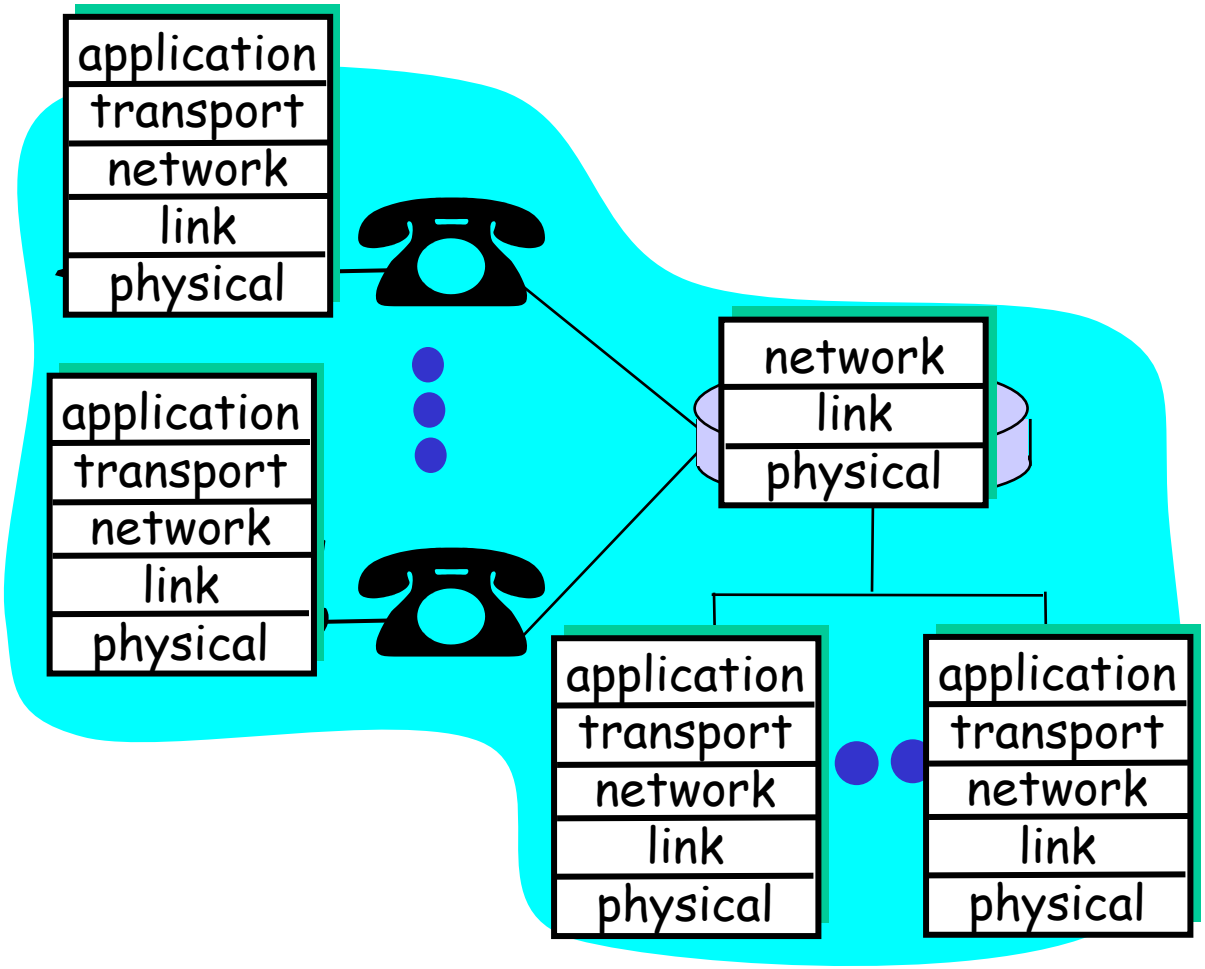
- **application:** supporting network applications
  - FTP, SMTP, STTP
- **transport:** host-host data transfer
  - TCP, UDP
- **network:** routing of datagrams from source to destination
  - IP, routing protocols
- **link:** data transfer between neighboring network elements
  - PPP, Ethernet
- **physical:** bits "on the wire"



Layering: logical communication

Each layer:

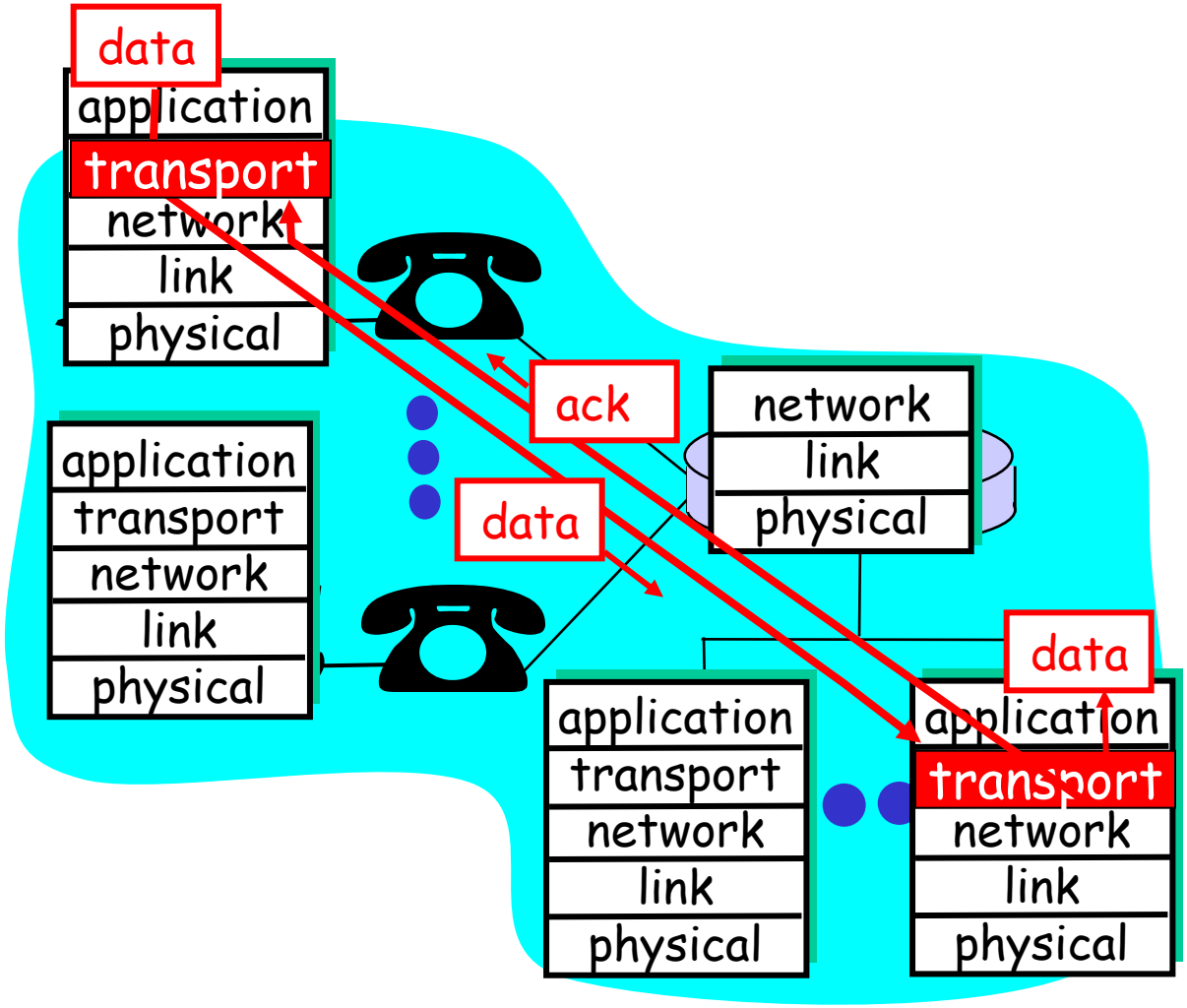
- distributed
- "entities" implement layer functions at each node
- entities perform actions, exchange messages with peers



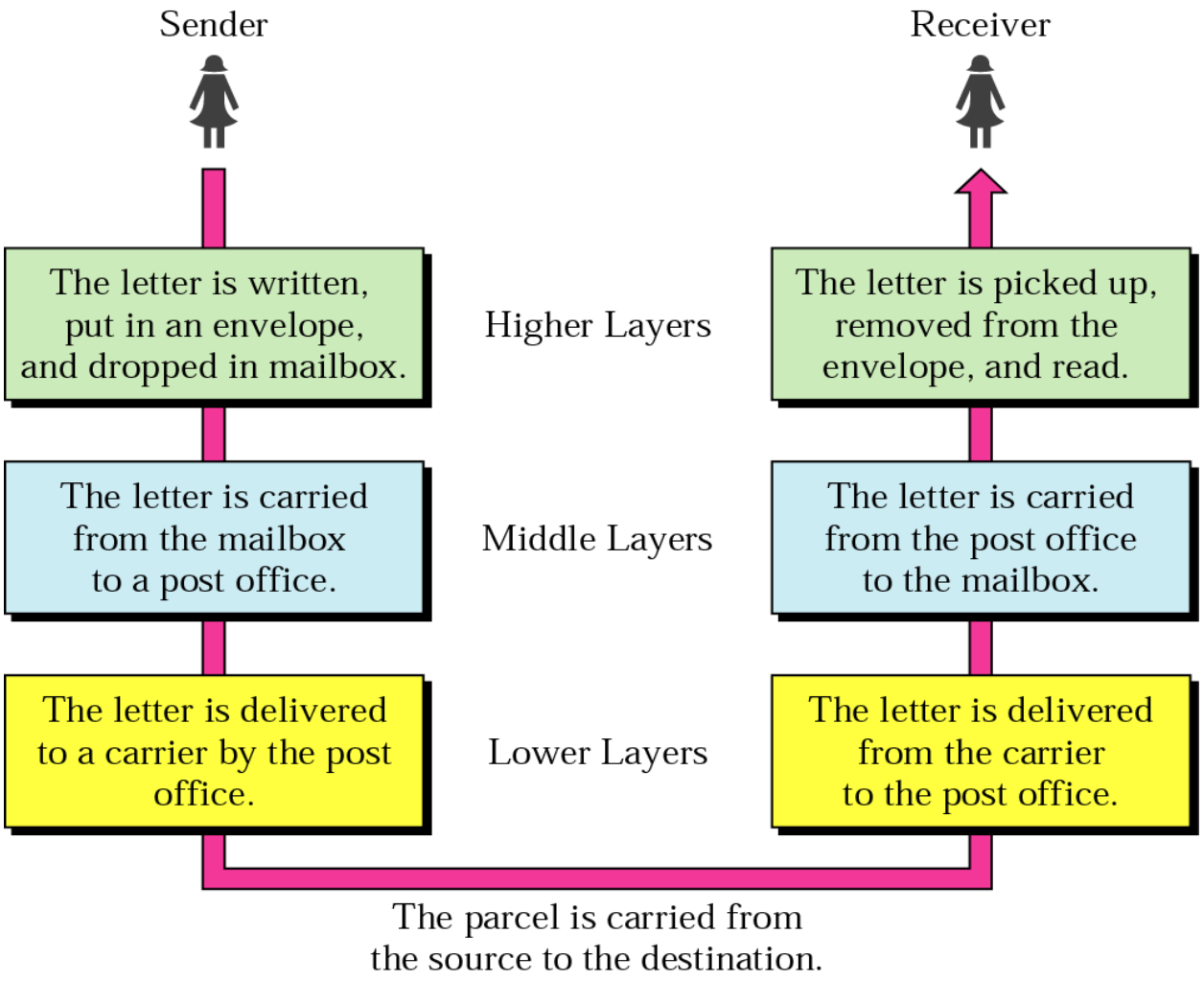
Layering: *logical* communication

E.g.: transport

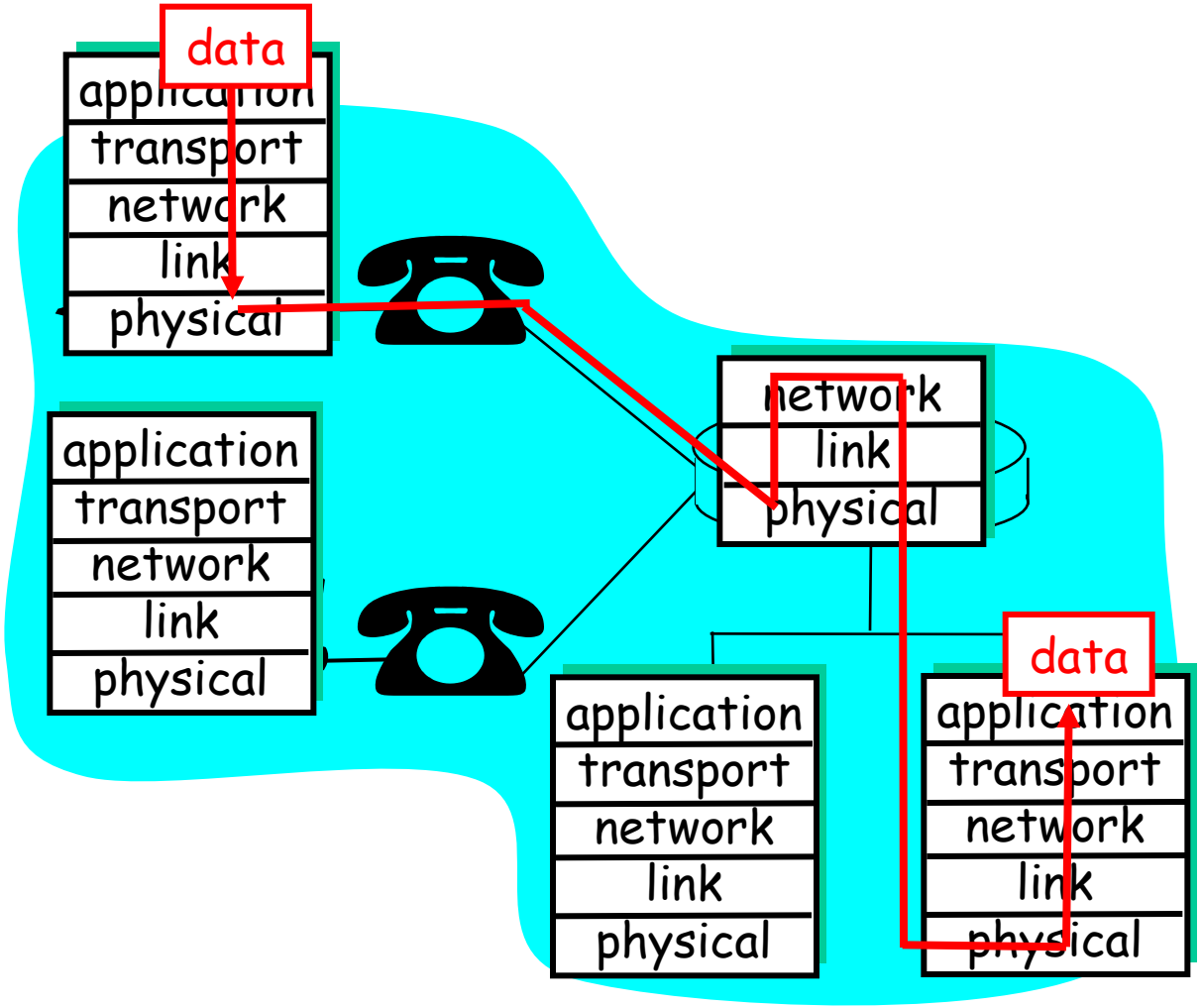
- take data from app
- add addressing, reliability check info to form "datagram"
- send datagram to peer
- wait for peer to ack receipt
- analogy: post office



post office analogy of layers



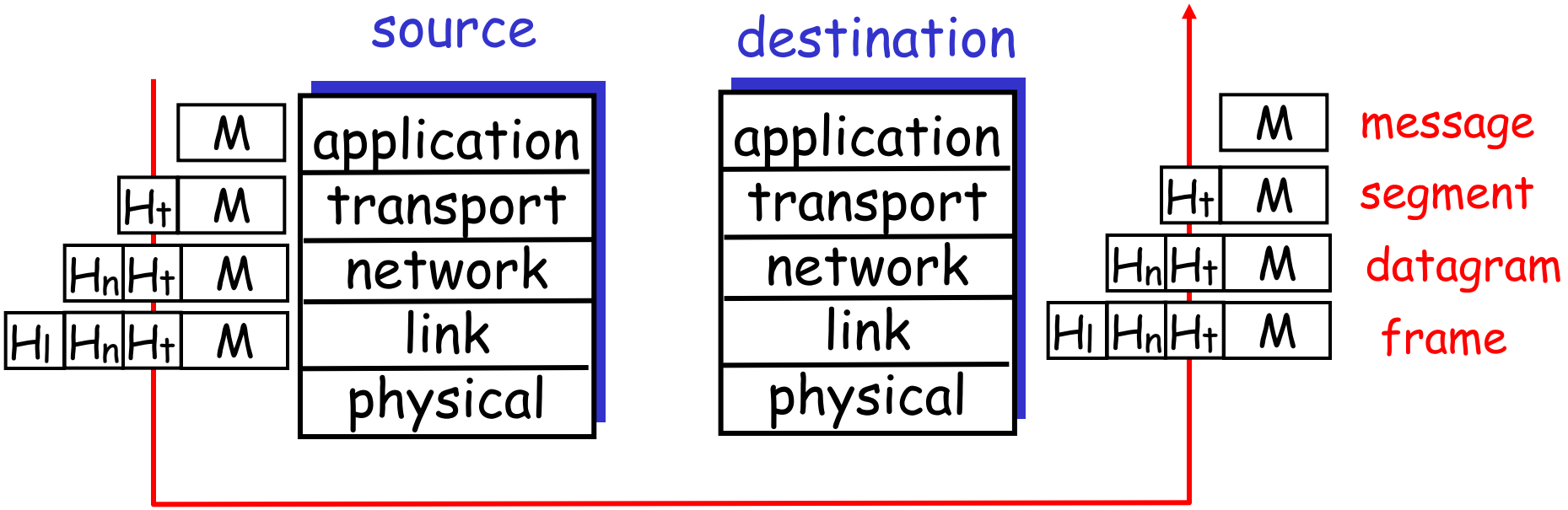
Layering: physical communication



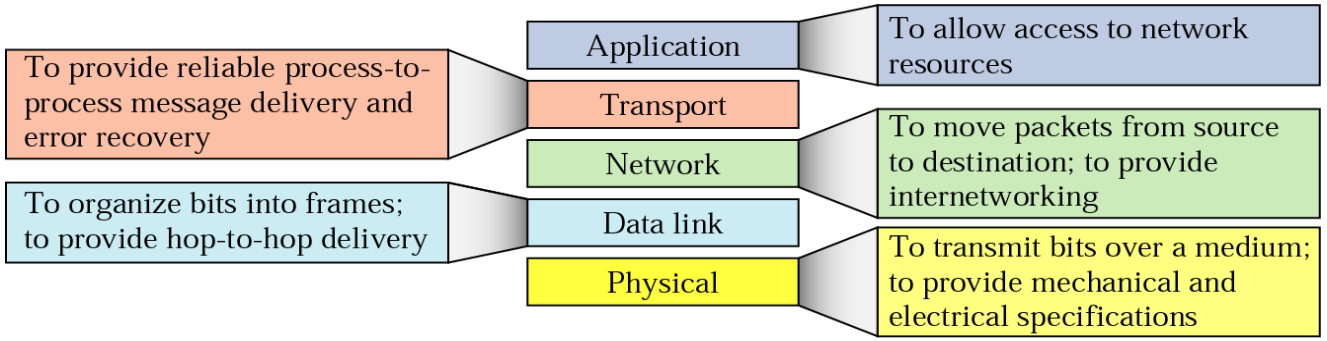
Protocol layering and data

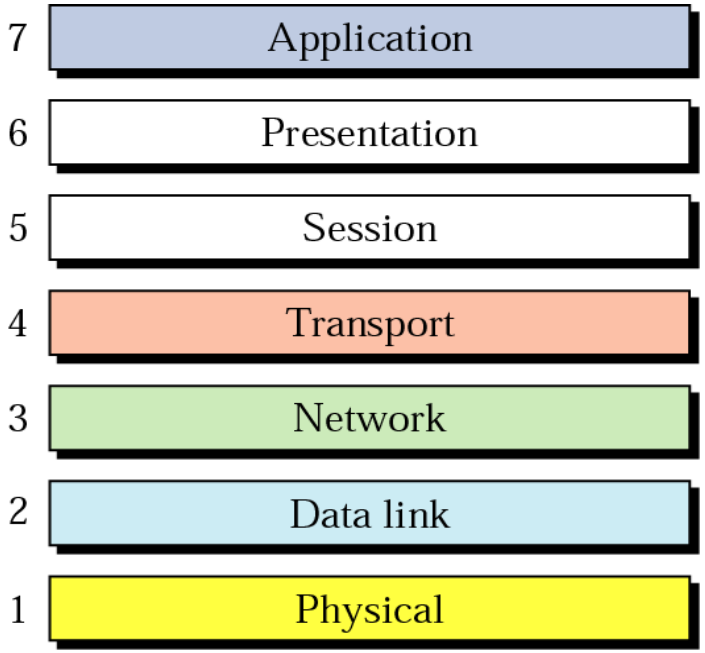
Each layer takes data from above

- adds header information to create new data unit
- passes new data unit to layer below



Summary of duties and OSI Model





Summary & Lecture Roadmap

- Covered a lot in today's lecture!
- We 'll discuss these (and more) in details during this course
- Internet overview
- what's a protocol?
- network edge, core, access network
  - packet-switching versus circuit-switching
- Internet/ISP structure
- performance: loss, delay
- layering and service models

Roadmap: what we'll learn in this course

- **Week 3: Internetworking, IPv4, IPv6**
- **Week 4: Physical Aspects of Data Communications, Data Encoding**
- **Week 5: Error Detection**
- **Week 6: Multiplexing**
- **Week 7: Flow Control, Error Control**
- **Week 8: Routing**
- **Week 9: LAN and Medium Access Control (MAC) Methods**
- **Week 10: Wireless Networking**
- **Week 11: Wide Area Networks, Packet and Circuit Switching, ATMs**
- **Week 12: Transport Protocols**
- **Week 13: Emerging Networking Technologies**

Internet History

1961-1972: Early packet-switching principles

- 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational

- 1972:
  - ARPAnet demonstrated publicly
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes

Internet History

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 1973: Metcalfe's PhD thesis proposes Ethernet
- 1974: Cerf and Kahn - architecture for interconnecting networks
- late70's: proprietary architectures: DECnet, SNA, XNA
- late 70's: switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

Cerf and Kahn's internetworking principles:

- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

define today's Internet architecture

Internet History

1980-1990: new protocols, a proliferation of networks

- 1983: deployment of TCP/IP
- 1982: SMTP e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: FTP protocol defined
- 1988: TCP congestion control

- new national networks: Csnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

Internet History

1990, 2000's: commercialization, the Web, new apps

- Early 1990's: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990's: commercialization of the Web

Late 1990's – 2000's:

- more killer apps: instant messaging, peer2peer file sharing (e.g., Naptser)
- network security to forefront
- est. 50 million host, 100 million+ users
- backbone links running at Gbps

webACE – world's tiniest webserver